plus one SlotTime before decrementation in the DCF. In the latter case, if the medium becomes busy again during this slot time, the backoff timer remains unchanged. Therefore, to accurately implement the EDCA functionality, the backoff timer needs to be changed in ns-3 simulator. The modification steps are as follows: First, the idle duration of medium is tracked by each station.

Once the idle duration of one station becomes greater than or equal to AIFS, the next step is triggered. In step two, the slot boundary of AIFS is identified and the potential decrement for the backoff timer counter is increased by one. Hence, the length of the idle duration after AIFS is divided by the length of one SlotTime and, in step three, the backoff timer counter is decremented by the quotient plus one. When the backoff timer reaches zero, the station is allowed to immediately transmit after the medium becomes idle for one AIFS.

Additional Backoff Timer Trigger

EDCA-mac-fixes-patches.tar.gz

In ns-3, once a frame is generated in the upper layer and ready for transmission, the station requests to access to the medium immediately. According to the standard, there are two possible situations for this access request.

- If the medium is busy as indicated by either physical or virtual carrier sensing, and the counter of backoff timer is equal to zero, the station should invoke a backoff procedure.
- If the counter of backoff timer is equal to zero and the medium is idle for at least one AIFS (i.e., the medium is not seized by another station), the station should immediately start the transmission.

Taking into consideration the situation where the counter of backoff timer is zero and the idle duration is less than one AIFS. There are two options: after AIFS, the station either transmits the frame immediately, or initiates an additional backoff procedure. Based on the conditions for initiating a transmission for EDCA in the 802.11 standard: “there is a frame available for transmission at that Enhanced Distributed Channel Access Function (EDCAF), and the backoff timer for that EDCAF has a value of 0”. Although the situation fits the condition, the second option is chosen, since even though the medium is indicated as idle when the station claims its request, the medium might not be truly idle. In other words, the AIFS is a part of the EDCA function that

cannot detect if another station has a packet ready for transmission. As an example, consider following case:

Both stations A and B have a value of 0 for the backoff timer counter and each station has one frame available for transmission. They remain quiet during the AIFS. However, after the AIFS both stations transmit the frame and a collision occurs. Thus, if the stations were allowed to send the frames generated during the AIFS directly, more collisions could have happened among stations. The solution to avoid these collisions is to initiate the backoff procedure for these frames. In the default ns-3, the frame can be sent out directly without the additional backoff procedure. Note that the standard leaves room for interpretation on this matter. Our interpretation in consultation with experts in the 802.11 standard activities yields to the most likely implementation in the field radio chips. Based on this feedback, we modified the channel-granting function to make sure the additional backoff procedure is implemented.

The modification is also made in class dcf-manager.cc. Recall that once a frame is generated from the upper layer and available for transmission, the `DcfManager::RequestAccess()` function is called to attempt to access the medium. In this function, both the backoff timer counter and the medium state are checked. If the backoff timer counter equals 0 and the medium is busy, a backoff procedure will be initiated. If the backoff timer counter has a value of 0 but the medium is idle, the length of this idle duration will be calculated. If the length is less than one AIFS (i.e., the station is requesting the access during AIFS) an additional backoff procedure will be started. Except above cases, new initialization of a backoff procedure is not necessary. Further actions are taken to update the backoff timer counter or grant the medium access.

There are three cases to initiate a new backoff timer:

- A channel access request is received when the channel is indicated as busy and the current backoff timer counter is 0
- A channel access request is received during the period of an ack timeout whenever it applies (the channel is conceptually considered as busy) and the current backoff timer counter is 0
- The channel access request is received during an IFS interval (i.e. AIFS or EIFS) and the current backoff timer counter is 0

The default ns3 handles the first case by calling a function called `DcaTxop::NotifyCollision()`. The name of this function is a bit confusing as this is not a collision. In addition, ns3 does not handle the second case properly. Based on the NAV vector which is set for other terminals listening to the transmitter or receiver, the ack timeout should be considered as part of a successful transmission. Therefore, the MAC layer should treat a transmission request received during an ack timeout just like the case it is received during another transmission. In this case, the device does not call for setting a backoff timer, instead, it just wait until the ack timeout passes. The third case is newly added.

All three cases are handled with a function named `DcfState::NotifyBackoffStart()`. This function is similar to `DcaTxop::NotifyCollision()`, but to add readability to the code, it has been renamed to reflect the more general use cases.

Queue Management Policy

wifi-mac-queue-patches.tar.gz

The default ns-3.16 drops the newest packet whenever the queue is full in the MAC layer. Some applications need to preserve the newest generated packet and drop the oldest one from the MAC layer queue. This patch has been developed to add this feature to ns-3 MAC layer queueing policy.

A second queue management policy was added to `WifiMacQueue::Enqueue()`. Before a new packet can be inserted into the queue, the function checks whether the queue is full or not. If the queue is full and the drop newest policy is selected, the newly generated packet will be dropped (as default ns-3 behaves). However, if the drop oldest policy is selected, the oldest packet in the queue will be dropped and the new packet will be inserted at the tail of the queue. The drop-oldest policy was added to transmit the latest generated packet with most recent information instead of an old packet.

Physical Layer: Frame Capture Effect Implementation

frame-capture-patches.tar.gz